REMARKS

This Application has been carefully reviewed in light of the Office Action mailed on March 25, 2005. Claims 1-39 are pending in the Application. Claims 1-39 were rejected in this Office Action. Claim 35 has been amended. Applicants respectfully request reconsideration and favorable action in this case.

Rejections Under 35 U.S.C. § 102(b):

Claims 1, 2, 5-10, 35, and 36 are rejected under 35 U.S.C. 102(b) as being anticipated by U.S. Patent No. 5,557,742 to Smaha et al. ("*Smaha*"). Applicants respectfully traverse.

Claim 1

Applicants' independent Claim 1 recites "generating, for each of the one or more signature definitions, an inspector instance based on the data file; and executing, for each of the one or more signature definitions, the generated inspector instance to detect network traffic matching the signature definition." The Office Action relies on *Smaha* column 10, lines 10-45 to teach this limitation, however this reliance is misplaced. This passage of *Smaha* details the operation of the misuse engine with respect to seeking a match for a signature data structure. However, no where in this passage are inspector instances generated for each signature definition. This passage also does not teach inspector instances that are executed to detect network traffic matching the signature definition. Rather, the signatures in *Smaha* are loaded and processed by the misuse engine without the creation of any signature instances that are executable. *See Smaha* column 9, lines 20-31. For at least this reason, Claim 1 is allowable as are Claims 2-10, which depend therefrom. Favorable action is requested.

Claim 35

Applicant has amended claim 35. Claim 35, as amended, should be allowed for analogous reasons to those discussed below for Claim 36.

Claim 36

Applicants' independent Claim 36 recites "a signature definition . . . comprising: an identifier for the signature; and one or more parameter-value pairs associated with the signature, each parameter-value pair comprising a parameter name and associated parameter value." The Office Action relies on *Smaha* to teach this limitation (column 8, lines 8-36), but this reliance is misplaced. This passage in *Smaha* states:

> FIG. 3 describes process flow 100 for operation of the data structure load mechanism and the elements of the data structure according to the present embodiment. In process flow 100, load mechanism 102 receives selectable misuse data from computer memory device 104 and from storage device 106. Relating FIG. 3 to FIG. 1, computer memory 104 may be thought of as computer memory 26. Storage device 28 and load mechanism 102 may be thought of as part of input mechanism 20 for selectable misuses. From these inputs, load mechanism 102 creates signature data structure 108. Load mechanism 102 loads the misuse elements to signature data structure 108 and creates index 110. Signature data structure 108 may be loaded from a predefined area of computer memory device 106 or from a predefined stored image that storage device 104 holds. The predefined stored images include stored representations of the remaining portions of signature data structure 108. A programmer or a misuse compiler or other software program could generate these images. Load mechanism 102 may also receive predefined elements from computer memory 106. This allows cooperating programs to control which misuses are available to misuse engine 30.
>
> Signature data structure 108 of the present embodiment contains the elements of a signature including index 110, initial state 112, transition functions 114, states 116, and end state 118. Taken together elements 112, 114, 116 and 118 embody a computer representation of a misuse, herein referred to as signature data structure 108. The computer representation of a misuse is created by a program or programmer from descriptions of misuses.

The Office Action states that it "considers signature data structure as applicants' signature definition and elements of signature data structure as applicants' identifiers, parameters-value pairs, parameter name and associated parameter value." However,

this is incorrect. This passage of *Smaha* does not teach signatures with parameter value pairs comprising a parameter name and associated parameter value. The signature data structure in *Smaha* lists only these elements: an index, initial state, transition functions, states, and an end state. No parameter-value pairs are taught by *Smaha*. For at least this reason, Claim 36 is allowable as are Claims 37-39, which depend therefrom.

Rejections Under 35 U.S.C. § 103(a):

Claims 11 and 19 have been rejected under 35 U.S.C. 103(a) over *Smaha* in view of U.S. Patent No. 5,557,742 to Allen Gluck et al. ("*Gluck*"). Claim 28 has been rejected under 35 U.S.C. 103(a) over *Smaha* in view of U.S. Patent No. 6,321,338 to Phillip A. Porras et al. ("*Porras*"). Applicants respectfully traverse.

Claim 11

Applicants' independent Claim 11 recites "automatically generating, for each of the one or more signatures defined in the default signature file, executable code operable to detect intrusions associated with the default signature." The Office Action concedes that *Smaha* does not teach this limitation. The Office Action relies on *Gluck* (column 7, lines 45-59) to teach this limitation, but this reliance is misplaced. The passage of *Gluck* relied upon by the Office action states:

> At step 230, the processor prompts the user if an auto update is desired. In this prompted moded which is intended for used by less sophisticated users, a method of updating the virus signature files with a minimum number of decisions is provided. A full function mode, which is intended to be used by sophisticated users, provides a menu driven method enabling a user to select (1) different functions of the update program and (2) the order of execution of the selected functions.

In the system and method taught by *Gluck,* a user must answer prompts and use a menu driven method to update signatures. Therefore, the signature update taught by *Gluck* is clearly not *automatic*, as Applicants' Claim 11 teaches with respect to code generation. Furthermore, the signatures in *Gluck* that are manually updated are not executable code, but are rather inputs into an anti-virus program. (See column 2, lines 47-52). For at least this reason, Claim 11 is allowable as are Claims 12-18, which depend therefrom.

Claim 19

Applicants' independent Claim 19 teaches "communicating to the sensor a desire to create a modified signature from a signature to be modified; receiving from the sensor data indicative of parameters and associated values for the signature to be modified; and providing to the sensor a modified value for at least one of the parameters to create a modified signature." The Office Action relies on *Gluck* (column 8, lines 20-49) to teach this limitation, but this reliance is misplaced. The passage of *Gluck* relied upon by the Office action states:

> In step 310, the processor 20 of the computer compares the versions of virus signature update files on the medium against the virus signature files 110 on the system 10. If all of the update virus signature files on the storage media are the same or older than that stored in the virus signature files 110 on the system 10 the processor 20 proceeds to step 360 where it carries on routine operations of associated with the system 10.
>
> However, if in step 310 the processor determines that at least one of the virus signature update files is not found in the virus signature files 110 of the system 10, the processor 20 proceeds to step 320 where it prompts the user to decide whether to update the virus signature files 110. If the user decides no, the processor 20 advances to step 360 where it carries on routine operations of associated with the system 10. If the user decides that he/she does wish to update the virus signature files 110, the processor 20 proceeds to step 330 where it copies the old versions of the virus signature files 110 to a temporary buffer (to save for use in case the update is not completed properly. Then in step 340, the processor 20 adds the virus signature updates to the virus signature files 110. In this step, duplicate virus signature files are overwritten with the updated versions and the virus signature updates that are new to the system are added to the virus signature files 110.
>
> The entire virus signature file 110 is not completely overwritten with the update virus signature files since the storage medium 170 might only contain new virus signatures rather than a comprehensive list of all virus signatures.

This passage of *Gluck* refers to a process and steps for updating virus signature files by overwriting the existing signatures. No modification of a signature is taught by

Gluck, nor are any of the elements for doing so taught. For instance, *Gluck* does not teach receiving from the sensor data indicative of parameters and associated values for the signature to be modified, nor does *Gluck* teach providing to a sensor a modified value for a least one of the parameters to create a modified signature. In *Gluck*, entire signatures are replaced. (See column 8, lines 42-43). Therefore, *Gluck* does not teach changing a value for a parameter of a signature in order to create a modified signature as is recited by claim 19 of Applicants' invention. For at least this reason, Claim 19 is allowable as are Claims 20-27, which depend therefrom.

Claim 28

Applicants' Claim 28 recites "network detection engine . . . operable to generate executable code based on either one of the stored default signatures or one of the stored user-defined signatures, the executable code operable to detect a network intrusion defined by the associated user-defined signature or the associated default signature." The Office Action relies on *Smaha* (column 10, lines 1-54) to teach this limitation, but this reliance is misplaced. This passage from *Smaha* details the operation of the misuse engine with respect to seeking a match for a signature data structure. (See column 9, line 20-31). The misuse engine searches for matches using signatures as inputs. However, nowhere in *Smaha* does it teach the misuse engine generating executable code based on the input signatures. The misuse engine itself in *Smaha* attempts to detect intrusions. However, in Claim 28 of Applicants' invention, the executable code generated by the network detection engine is operable to detect a network intrusion. For at least this reason, Claim 28 is allowable as are Claims 29-34, which depend therefrom.

CONCLUSION

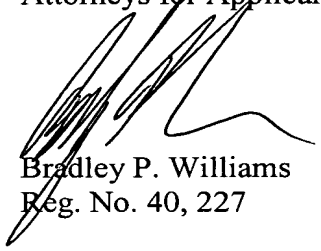
Applicants have now made an earnest attempt to place this case in condition for immediate allowance. For the foregoing reasons and for other apparent reasons, Applicants respectfully request full allowance of all pending Claims.

If the Examiner feels that a telephone conference or an interview would advance prosecution of this Application in any manner, please feel free to contact the undersigned attorney for Applicants.

Applicants do not believe that any fees are due. However, the Commissioner is hereby authorized to charge any fees or credit any overpayments to Deposit Account No. 02-0384 of Baker Botts L.L.P.

Respectfully submitted,

BAKER BOTTS L.L.P.
Attorneys for Applicant



Bradley P. Williams
Reg. No. 40, 227

Date: June 25, 2005

Correspondence Address:
Customer Number: **05073**